



## **Yape: Da Bi-Modalidade à Autonomia Celular**

*Arquitetura Distribuída, Governança de Dados e Resiliência Sistêmica na Modernização de  
Plataformas FinTech em Mercados Emergentes*

**Carlos Ulisses Flores**

**AMERICAN GLOBAL TECH UNIVERSITY**

**MESTRADO EM INTELIGÊNCIA ARTIFICIAL**

**MIT-507 — Digital Organization da American Global Tech University - AGTU.**

**Professor: José Eduardo Campos**

**15 de fevereiro de 2026**

## Resumo

Este estudo de caso analisa a evolução arquitetural do Yape, a plataforma líder de pagamentos digitais do Peru com mais de 16 milhões de usuários, diagnosticando as limitações de escalabilidade impostas pelo modelo de “TI Bi-Modal” e sua dependência de sistemas legados em Mainframe — o fenômeno denominado “Hollow Core”. Para mitigar riscos sistêmicos e habilitar crescimento sustentável, propõe-se a transição para uma Arquitetura Baseada em Células (Cell-Based Architecture), integrada a princípios de Zero Trust, governança de dados via Data Mesh e gestão de riscos de IA alinhada ao NIST AI RMF 1.0. A eficácia da proposta foi validada através de uma Simulação de Eventos Discretos (DES) estocástica em Python/SimPy, cujos resultados demonstram que, sob estresse de 2.000 transações por segundo, o modelo celular mantém latência percebida determinística e elimina o colapso por saturação de filas. Conclui-se que o desacoplamento do ledger transacional constitui um imperativo estratégico para garantir continuidade operacional, confiança do consumidor e prontidão para Inteligência Artificial auditável.

**Palavras-chave:** Arquitetura Baseada em Células; Simulação de Eventos Discretos; Resiliência Sistêmica; FinTech; Data Mesh; Zero Trust.

## **Abstract**

This case study analyzes the architectural evolution of Yape, Peru's leading digital payment platform with over 16 million users, diagnosing scalability constraints imposed by the "Bi-Modal IT" model and its Mainframe legacy dependency — the "Hollow Core" phenomenon. To mitigate systemic risks and enable sustainable growth, a transition to Cell-Based Architecture is proposed, integrated with Zero Trust security, Data Mesh governance, and AI risk management aligned with NIST AI RMF 1.0. The proposal's efficacy was validated through a stochastic Discrete Event Simulation (DES) in Python/SimPy. Results demonstrate that under 2,000 TPS stress the cellular model maintains deterministic user-perceived latency and eliminates queue saturation collapse, outperforming the monolithic model in resilience, throughput, and SLA compliance. Decoupling the transactional ledger constitutes a strategic imperative for operational continuity, consumer trust, and readiness for auditable AI-driven credit scoring.

**Keywords:** Cell-Based Architecture; Discrete Event Simulation; Systemic Resilience; FinTech; Data Mesh; Zero Trust.

# 1. Introdução

No cenário contemporâneo de serviços financeiros, a latência transcendeu sua definição puramente métrica para tornar-se um determinante direto de competitividade, confiança do consumidor e viabilidade econômica (Verbitski et al., 2017). A ascensão dos pagamentos instantâneos exige que as instituições bancárias transcendam as limitações das infraestruturas legadas. A economia always-on não tolera janelas de manutenção, downtime programado ou inconsistência de dados.

Neste contexto, o Yape, a plataforma de pagamentos digitais incubada pelo Banco de Crédito del Perú (BCP), emerge como um estudo de caso paradigmático de hipercrecimento confrontado por barreiras arquiteturais, conforme documentado no caso de transformação digital do BCP (McKinsey, 2023). Com mais de 16 milhões de usuários e 13 milhões de usuários ativos mensais (TerraPay, 2024), o Yape opera como principal vetor de inclusão financeira no Peru. Dados do Banco Mundial indicam que aproximadamente 43% dos adultos peruanos permaneciam sem conta bancária formal em 2021 (World Bank, 2022), cenário no qual o Yape passou a servir cerca de 5 milhões de usuários previamente não bancarizados (TerraPay, 2024). Contudo, sua arquitetura atual enfrenta o desafio crítico do “Hollow Core” (Núcleo Oco): uma dissonância operacional onde canais digitais de alta velocidade colidem com um núcleo de liquidação baseado em Mainframes rígidos.

Esta dependência cria gargalos de escalabilidade não lineares, riscos sistêmicos de disponibilidade e um custo de fricção operacional que ameaça a sustentabilidade do modelo de negócios. À medida que o volume transacional cresce exponencialmente, o modelo monolítico exhibe sinais de fadiga estrutural, manifestados em latências imprevisíveis durante picos de demanda.

O presente estudo tem como objetivo diagnosticar as vulnerabilidades deste modelo bi-modal e propor uma migração estratégica para uma Arquitetura Baseada em Células. Para transcender

a análise puramente teórica, a viabilidade da proposta é validada através de uma Simulação de Eventos Discretos (DES). A análise estrutura-se em quatro eixos: (i) diagnóstico do acoplamento legado e do fenômeno Hollow Core; (ii) especificação da arquitetura celular com shuffle sharding; (iii) governança de dados via Data Mesh e gestão de riscos de IA sob o NIST AI RMF 1.0 (Tabassi, 2023); e (iv) evidência empírica via modelos estocásticos rigorosos. O código-fonte e os artefatos de análise estão disponíveis publicamente no repositório GitHub do autor (Flores, 2026).

## **2. Desenvolvimento**

### **2.1. Artigos Selecionados e Fundamentação Teórica**

A construção do arcabouço teórico fundamenta-se em três pilares de literatura contemporânea que cobrem arquitetura distribuída, governança de dados e regulação de IA.

**Pilar 1 — Arquitetura de Resiliência em Hiperescala.** A documentação da AWS sobre Arquiteturas Baseadas em Células (2024) fornece a base para isolamento de falhas via bulkheads e shuffle sharding. Complementarmente, Taft et al. (2020), no ACM SIGMOD, detalham o CockroachDB como banco SQL distribuído geo-replicado com consistência ACID forte. A problemática da latência de cauda em sistemas distribuídos, analisada seminalmente por Dean e Barroso (2013) nas ACM Communications, fundamenta a necessidade de arquiteturas que limitem o blast radius: em sistemas com milhares de componentes, mesmo probabilidades baixas de latência elevada em componentes individuais produzem degradação agregada significativa. A escolha destas fontes justifica-se pela conexão direta com o problema: o caso Yape necessita de isolamento de falhas (células) e consistência transacional global (NewSQL) para superar o acoplamento ao mainframe.

**Pilar 2 — Governança de Dados Descentralizada.** Dehghani (2022) introduz o paradigma Data Mesh com quatro princípios: propriedade de domínio, dados como produto, plataforma self-service e governança computacional federada. Para o Yape, este framework resolve o paradoxo criado pela arquitetura celular: enquanto as células isolam dados para resiliência, o negócio exige visibilidade analítica transversal para detecção de fraudes e credit scoring.

**Pilar 3 — IA Responsável e Segurança.** O NIST AI RMF 1.0 (Tabassi, 2023) estabelece as funções de Governar, Mapear, Medir e Gerenciar riscos de IA, fornecendo estrutura para mitigar vieses em credit scoring de populações não bancarizadas. A Estratégia Zero Trust do DoD (2022) complementa o pilar de segurança, estabelecendo que a identidade — e não o perímetro — deve fundamentar todo controle de acesso em ambientes distribuídos.

## **2.2. Análise Técnica e Arquitetural**

### ***2.2.1. Diagnóstico AS-IS: O Dilema do “Hollow Core”***

A infraestrutura atual do Yape implementa TI Bi-Modal: na camada de frontend, microsserviços orquestrados via Kubernetes (Amazon EKS) com CI/CD avançado e Infrastructure as Code (IaC) via Terraform para provisionamento imutável; na camada de persistência, porém, a liquidação final permanece acoplada ao Core Banking legado em Mainframes IBM z/OS. Essa dicotomia estabelece a separação clássica entre processamento transacional (OLTP) na nuvem e processamento analítico (OLAP) dependente de extrações batch do mainframe, impedindo a operação analítica em tempo real.

Para mitigar a latência inerente ao mainframe, o Yape implementou um Shadow Ledger utilizando AWS Aurora PostgreSQL (Verbitski et al., 2017). Embora permita pagamentos em tempo quase real, esta estratégia introduz risco sistêmico de consistência eventual: o saldo visualizado pelo usuário é uma “promessa de estado” que só se torna imutável após

sincronização assíncrona com o System of Record no mainframe. Economicamente, a reconciliação constante entre Shadow Ledger e Core Ledger consome recursos computacionais e gera indisponibilidade durante janelas de batch noturnas.

### **2.2.2. Arquitetura TO-BE: Cell-Based Architecture e Ledger Distribuído**

A documentação AWS define cada célula como uma instância completa e independente da stack, dimensionada para um shard de usuários (AWS, 2024). O benefício central é a redução do Blast Radius: a falha de uma célula afeta estritamente seus usuários. Para o estado da arte em resiliência, propõe-se Shuffle Sharding na camada de roteamento — com 100 células e dependência de 2 por cliente, existem  $C(100,2) = 4.950$  combinações, reduzindo a probabilidade de impacto total a frações insignificantes (AWS, 2024). O provisionamento celular seria gerido via IaC (Terraform) com pipelines de Chaos Engineering para validação contínua de resiliência.

A proposta substitui o Shadow Ledger por CockroachDB, que utiliza consenso Raft para consistência ACID global, eliminando a reconciliação com o mainframe (Taft et al., 2020). O particionamento geográfico garante que dados de PII residam fisicamente no Peru para compliance regulatório. Com essa transição, a camada OLTP migra integralmente para o ledger distribuído, enquanto a camada OLAP opera sobre réplicas de leitura alimentando um Data Lakehouse (Apache Iceberg sobre S3), desacoplando definitivamente processamento transacional e analítico.

### **2.2.3. Governança de Dados: Data Mesh e Data Lakehouse**

A transição para células independentes introduz fragmentação de dados. O paradigma Data Mesh (Dehghani, 2022) resolve este dilema: (i) equipes de “Pagamentos QR” e “Microcrédito” assumem propriedade dos seus domínios; (ii) dados são tratados como produtos com SLAs, documentação e qualidade garantida; (iii) infraestrutura abstraída via plataforma self-service; e (iv) governança codificada (Policy as Code) com data lineage rastreado de ponta a ponta. O

Data Lakehouse complementa esta governança ao unificar a camada analítica com transações ACID sobre formatos abertos como Apache Iceberg, conforme o paradigma proposto por Armbrust et al. (2021) no CIDR, permitindo consultas OLAP cross-cell sem comprometer o isolamento transacional das células.

#### **2.2.4. IA Responsável e NIST AI RMF**

O NIST AI RMF 1.0 (Tabassi, 2023) fornece quatro funções — Governar, Mapear, Medir e Gerenciar — para mitigar riscos de viés em credit scoring. No Yape, onde 5 milhões de usuários não possuem relação bancária formal (TerraPay, 2024; World Bank, 2022), modelos treinados em dados históricos podem perpetuar exclusões financeiras. A arquitetura celular habilita: isolamento de dados de treinamento, monitoramento de deriva de modelo (model drift) em tempo real via pipelines MLOps (treinamento, validação, deploy e rollback automatizados), e rastreabilidade de decisões de crédito via imutabilidade do ledger distribuído. Para garantir transparência, propõe-se a adoção de Model Cards (Mitchell et al., 2019) como documentação padronizada de cada modelo de credit scoring, registrando métricas de desempenho desagregadas por segmento demográfico.

#### **2.2.5. Zero Trust, DevSecOps e Monitoramento Contínuo**

Em ambiente distribuído e celular, a segurança de perímetro tradicional torna-se obsoleta. Alinhada à Estratégia Zero Trust do DoD (2022), a arquitetura implementa mTLS (Mutual TLS) e identidade de carga de trabalho: cada interação entre microsserviços exige autenticação mútua criptográfica, impedindo movimentação lateral mesmo em caso de comprometimento de container. A prática de DevSecOps integra análise estática de segurança (SAST), varredura de dependências e testes de penetração automatizados nos pipelines de CI/CD, garantindo que vulnerabilidades sejam detectadas antes da promoção a produção. O monitoramento contínuo é operacionalizado via SIEM (Security Information and Event Management) correlacionando



eventos de autenticação, latência anômala e alertas de WAF em um painel unificado de observabilidade.

## **2.3. Discussão**

A análise comparativa dos modelos AS-IS e TO-BE revela que a arquitetura celular resolve simultaneamente múltiplas dimensões do problema: escalabilidade (particionamento horizontal), resiliência (bulkheads e shuffle sharding), consistência (NewSQL ACID distribuído), governança (Data Mesh federado com Data Lakehouse) e segurança (Zero Trust com DevSecOps). A integração destes cinco pilares é o diferencial crítico: a segurança protege os dados que fluem através das células governadas pelo Data Mesh, em infraestrutura provisionada por IaC e validada por Chaos Engineering, seguindo princípios de fluxo contínuo e limitação de trabalho em progresso (Anderson, 2010).

Contudo, a transição implica trade-offs significativos. O CockroachDB, embora garanta consistência ACID global, introduz latência adicional de consenso Raft em escritas multi-região — estimada entre 50–150 ms para commits inter-regionais (Taft et al., 2020) —, o que pode impactar transações de alta frequência se o particionamento geográfico não for cuidadosamente projetado. Adicionalmente, a complexidade operacional de um ambiente celular com Data Mesh exige maturidade organizacional elevada: equipes de domínio devem assumir responsabilidade end-to-end por seus data products, o que demanda investimento substancial em capacitação e cultura de dados (Dehghani, 2022).

A literatura de resiliência em hiperescala sustenta a proposta: o padrão de células com shuffle sharding, validado em produção pela AWS para serviços como Route 53 e DynamoDB (AWS, 2024), demonstra redução de blast radius a níveis estatisticamente insignificantes. A problemática da latência de cauda (Dean & Barroso, 2013) reforça este argumento: em

sistemas monolíticos compostos por centenas de microsserviços, a probabilidade de pelo menos um componente exibir latência elevada cresce exponencialmente com o número de dependências, tornando o isolamento celular não uma otimização, mas uma necessidade estrutural. No contexto FinTech peruano, essa resiliência é particularmente crítica dado o papel do Yape na inclusão financeira: uma indisponibilidade prolongada afeta desproporcionalmente populações que dependem exclusivamente desta plataforma. As melhorias integradas alinham cloud (células multi região com IaC), dados (Data Mesh e Lakehouse), IA (NIST AI RMF 1.0 com MLOps) e cibersegurança (Zero Trust com DevSecOps) sob uma perspectiva de sustentabilidade técnica e gestão de riscos.

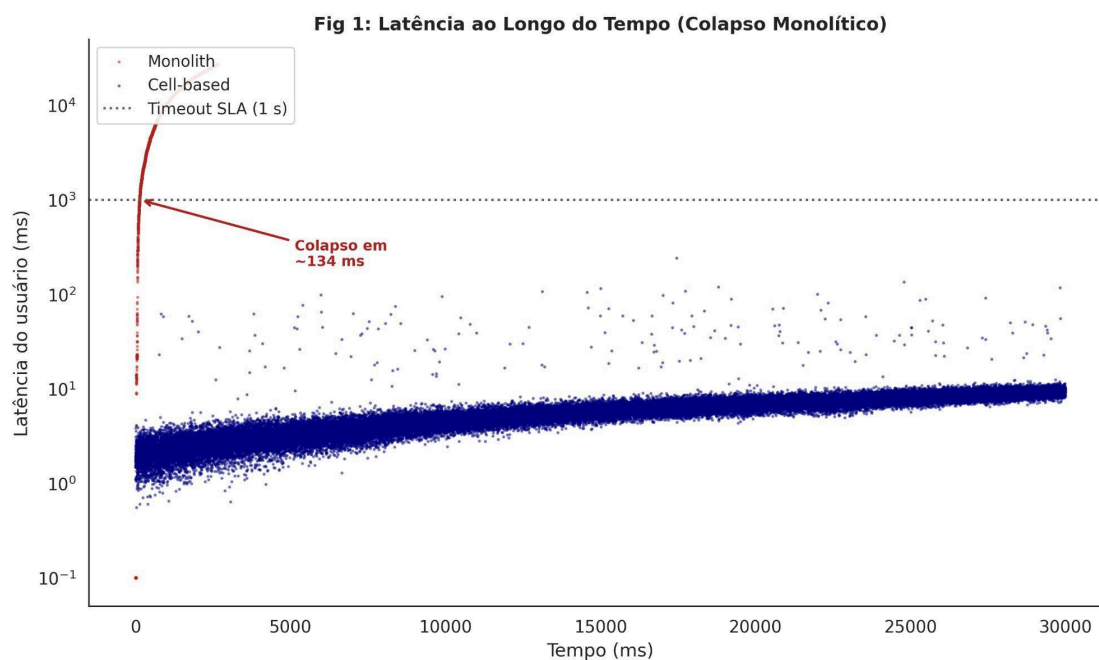
## 2.4. Validação Experimental: Simulação de Eventos Discretos

Para fornecer evidência empírica, conduziu-se uma Simulação de Eventos Discretos (DES) estocástica utilizando Python/SimPy, submetendo ambos os modelos a um cenário de estresse de 2.000 TPS durante 30 segundos simulados (Flores, 2026). O código-fonte, dados brutos e artefatos de proveniência estão no repositório GitHub do autor (<https://github.com/ulissesflores/mit507-yape-architecture-sim>, tag v1.0, commit fixado no snapshot Zenodo) com DOI permanente no Zenodo (DOI: 10.5281/zenodo.18645894), garantindo que a versão exata do código utilizada neste estudo seja recuperável independentemente de alterações futuras no repositório. Os parâmetros completos do experimento estão documentados no Anexo A e no arquivo `experiment_provenance.json` do repositório.

**Definição de métricas.** É fundamental distinguir duas métricas: (i) *latência percebida pelo usuário*, que corresponde ao tempo até o ACK (confirmação de recebimento da transação pela

API), e (ii) *latência de conclusão*, que abrange o tempo total até a finalização do processamento (settlement, verificação de fraude, persistência no ledger). Na arquitetura celular assíncrona, a latência percebida é desacoplada da latência de conclusão: o usuário recebe confirmação imediata (ACK) enquanto o processamento interno conclui de forma assíncrona. No modelo monolítico síncrono, ambas as métricas coincidem. Os gráficos a seguir reportam a latência percebida pelo usuário, que é a métrica crítica para SLA e experiência do consumidor.

**Latência ao longo do tempo.** A Figura 1 apresenta a latência percebida de cada transação em escala logarítmica. O colapso monolítico ocorre em duas fases: (i) saturação do pool de conexões e crescimento irreversível da fila a partir de  $t \approx 2.500$  ms, seguida de (ii) queda da vazão para  $\sim 0$  quando os timeouts acumulados passam a impedir a emissão de ACK ( $t \approx 3.000$  ms). A partir deste ponto, a latência escala indefinidamente, ultrapassando  $10^4$  ms. A arquitetura celular mantém latência estável na faixa de 2–10 ms durante toda a janela.



*Figura 1 — Latência Percebida ao Longo do Tempo: Colapso Monolítico vs. Estabilidade Celular (Flores, 2026).*

**Risco de Compliance (ECDF).** A Figura 2 exibe a Função de Distribuição Empírica Acumulada da latência percebida (ACK). Na arquitetura celular, 100% das transações registraram latência percebida abaixo do SLA de 1 segundo. No cenário monolítico síncrono, onde latência percebida coincide com latência de conclusão, apenas 5,4% cumpriram o requisito — violação sistemática que, em produção, acarretaria penalidades regulatórias.

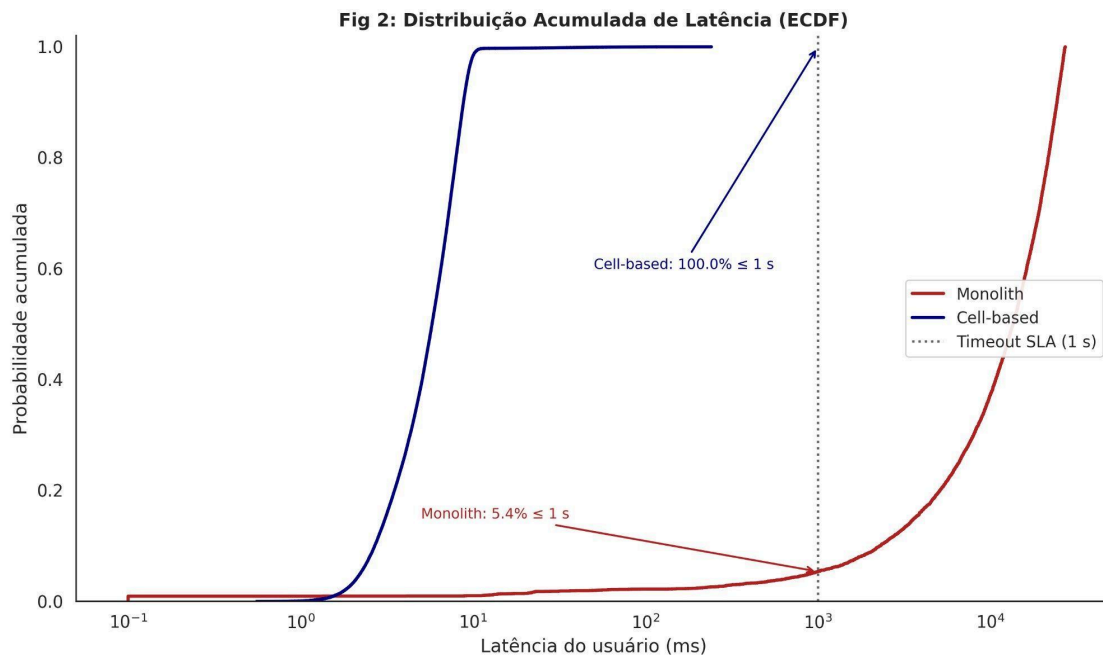
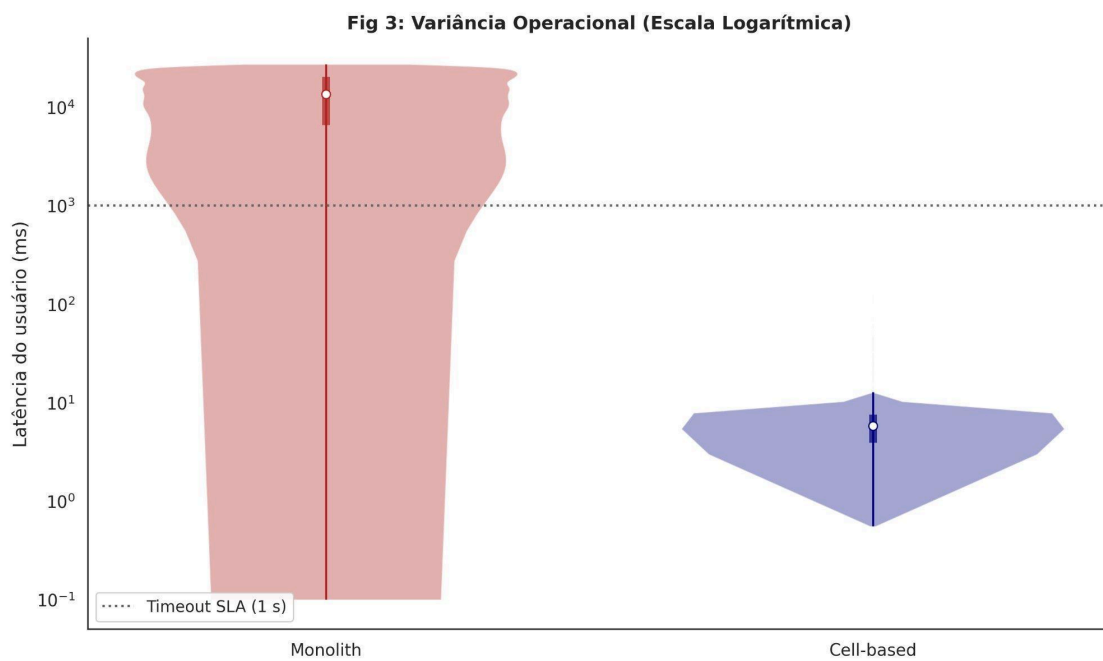


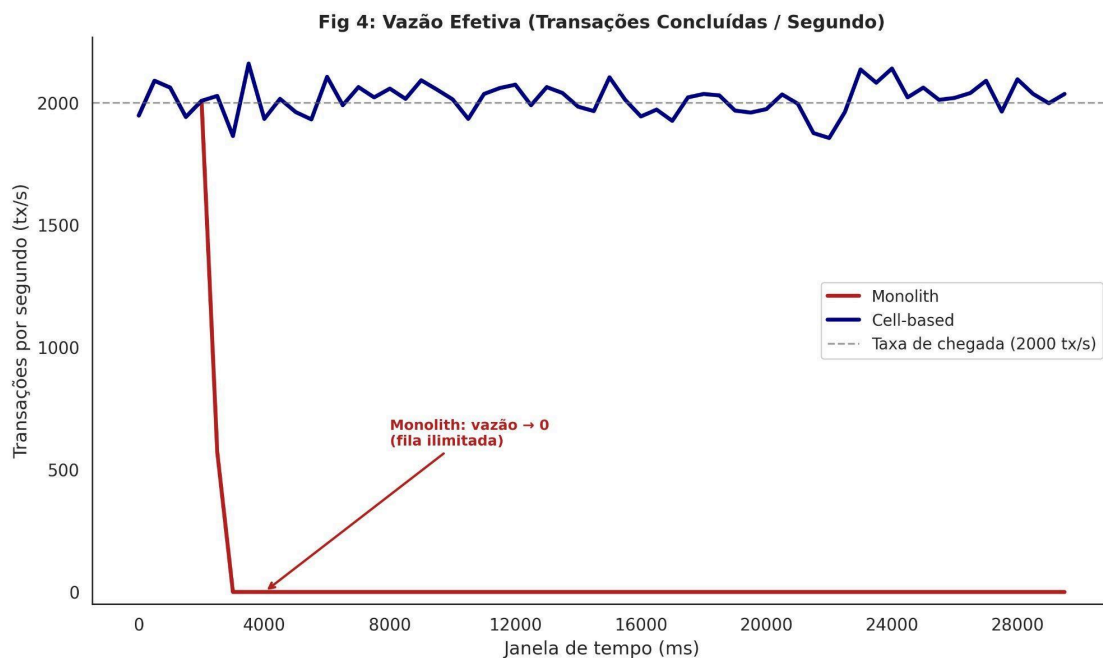
Figura 2 — Distribuição Acumulada de Latência (ECDF): Conformidade com SLA de 1s (Flores, 2026).

**Volatilidade Operacional.** A Figura 3 apresenta distribuição via violin plot em escala logarítmica. O monólito exibe variância de cinco ordens de magnitude ( $10^{-1}$  a  $\sim 2,7 \times 10^4$  ms), comportamento caótico que impossibilita planejamento de capacidade. A arquitetura celular concentra a densidade em faixa estreita (2–10 ms), inteiramente abaixo do SLA.



*Figura 3 — Variância Operacional: Violin Plot em Escala Logarítmica (Flores, 2026).*

**Vazão Efetiva (Throughput).** A Figura 4 demonstra a vazão efetiva (transações com ACK entregue por segundo) ao longo do tempo. Para efeitos de throughput, considera-se transação “concluída” no instante de emissão do ACK ao usuário. Após a segunda fase do colapso (~3.000 ms simulados), o monólito converge para zero transações processadas — fila ilimitada onde nenhum ACK é emitido dentro do timeout. A arquitetura celular sustenta ~2.000 TPS de forma estável, demonstrando escalabilidade horizontal.



*Figura 4 — Vazão Efetiva: Colapso Monolítico vs. Sustentação Celular a 2.000 TPS (Flores, 2026).*

### 3. Considerações Finais

O presente estudo demonstrou, através de diagnóstico técnico e validação experimental, que a arquitetura monolítica baseada em mainframes e shadow ledgers representa um risco

existencial para a escalabilidade do Yape. A simulação estocástica (Flores, 2026) confirmou que a Arquitetura Baseada em Células oferece resiliência superior: latência percebida (ACK) determinística, vazão linear e 100% de conformidade de SLA para latência percebida sob estresse de 2.000 TPS, contra apenas 5,4% no modelo monolítico.

Mais do que uma atualização tecnológica, a convergência de Data Mesh (Dehghani, 2022), Zero Trust com DevSecOps (DoD, 2022) e governança de IA alinhada ao NIST AI RMF (Tabassi, 2023) posiciona o Yape na fronteira da inovação fintech global. A descentralização do ledger via CockroachDB (Taft et al., 2020) elimina o ponto único de falha do Hollow Core, enquanto a governança federada com Data Lakehouse assegura conformidade regulatória e soberania de dados.

**Ameaças à validade.** A simulação utiliza dados sintéticos com semente aleatória fixa (seed=42) e execução única, o que é adequado para demonstrar relações causais em sistemas de filas (Little, 1961) mas não substitui testes de carga em produção. É importante notar que as métricas reportadas capturam a latência percebida pelo usuário (ACK), não a latência de conclusão end-to-end de settlement, que no modelo assíncrono inclui processamento posterior ao ACK. Estudos futuros devem incorporar múltiplas repetições com intervalos de confiança de 95%, análise de sensibilidade paramétrica (variando taxa de chegada e capacidade celular) e curvas de backlog para complementar a análise de latência de conclusão.

Os desafios contínuos residem na complexidade organizacional — que exige mudanças culturais profundas nas equipes de engenharia — e na preparação para ameaças emergentes, incluindo criptografia pós-quântica e regulação de IA generativa. Conclui-se que a transformação arquitetural proposta constitui o imperativo estratégico que permitirá ao Yape transcender suas limitações físicas, garantindo liderança na economia digital always-on com responsabilidade, auditabilidade e inclusão financeira como princípios norteadores.

## 4. Referências

- Amazon Web Services. (2024). Reducing the Scope of Impact with Cell-Based Architecture. AWS Well-Architected Framework. Disponível em: <https://docs.aws.amazon.com/wellarchitected/latest/reducing-scope-of-impact-with-cell-based-architecture/what-is-a-cell-based-architecture.html> (Acesso em: 10 de fevereiro de 2026).
- Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
- Armbrust, M., Ghodsi, A., Xin, R. & Zaharia, M. (2021). 'Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics'. Proceedings of the 11th Conference on Innovative Data Systems Research (CIDR '21).
- Dean, J. & Barroso, L. A. (2013). 'The Tail at Scale'. Communications of the ACM, 56(2), pp. 74–80. <https://doi.org/10.1145/2408776.2408794>
- Dehghani, Z. (2022). Data Mesh: Delivering Data-Driven Value at Scale. O'Reilly Media. ISBN: 9781492092391.
- Department of Defense (DoD). (2022). DoD Zero Trust Strategy. DoD CIO Zero Trust Portfolio Management Office. Disponível em: <https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf> (Acesso em: 10 de fevereiro de 2026).
- Flores, C. U. (2026). Cell-Based Architecture Simulation Artifacts: Stochastic Analysis of Transactional Latency. Zenodo. <https://doi.org/10.5281/zenodo.18645894>. Repositório: <https://github.com/ulissesflores/mit507-yape-architecture-sim>
- Little, J. D. C. (1961). 'A Proof for the Queuing Formula:  $L = \lambda W$ '. Operations Research, 9(3), pp. 383–387. <https://doi.org/10.1287/opre.9.3.383>



- McKinsey & Company. (2023). BCP: Transforming Peru's Banking Industry. Disponível em: <https://www.mckinsey.com/capabilities/tech-and-ai/how-we-help-clients/rewired-in-action/bcp-taking-banking-to-new-heights-on-a-digital-rocketship> (Acesso em: 10 de fevereiro de 2026).
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D. & Gebru, T. (2019). 'Model Cards for Model Reporting'. Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT\* '19), pp. 220–229. <https://doi.org/10.1145/3287560.3287596>
- Tabassi, E. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.AI.100-1>
- Taft, R., Sharif, I., Matei, A., VanBenschoten, N., Lewis, J., Grieger, T., Niemi, K., Woods, A., Birzin, A., Poss, R., Bardea, P., Ranade, A., Darnell, B., Gruneir, B., Jaffray, J., Zhang, L. & Mattis, P. (2020). 'CockroachDB: The Resilient Geo-Distributed SQL Database'. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1493–1509. <https://doi.org/10.1145/3318464.3386134>
- TerraPay. (2024). Yape Peru's Super App and TerraPay Join Forces to Accelerate Digital Inclusion. Press Release, 29 de outubro de 2024. Disponível em: <https://www.prnewswire.com/news-releases/yape-perus-supper-app-and-terrapay-join-forces-to-accelerate-digital-inclusion-302289565.html> (Acesso em: 10 de fevereiro de 2026).
- Verbitski, A., Gupta, A., Saha, D., Brahmadesam, M., Gupta, K., Mittal, R., Krishnamurthy, S., Maurice, S., Kharatishvili, T. & Bao, X. (2017). 'Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases'. SIGMOD '17, pp. 1041–1052. <https://doi.org/10.1145/3035918.3056101>

World Bank. (2022). The Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of COVID-19. Washington, DC: World Bank.

<https://doi.org/10.1596/978-1-4648-1897-4>

## Anexo A — Parâmetros da Simulação DES

A Tabela 1 apresenta os parâmetros completos de proveniência experimental utilizados na Simulação de Eventos Discretos, extraídos do arquivo `experiment_provenance.json` arquivado no repositório (Flores, 2026).

**Tabela 1 — Parâmetros de Proveniência Experimental DES (Flores, 2026).**

Parâmetro	Monólito (A)	Cell-Based (B)
Modo	MONOLITH	CELL_BASED
Tipo de tráfego	STRESS	STRESS
Semente aleatória	42	42
Tempo de simulação (ms)	30.000	30.000
Taxa de chegada normal (TPS)	50	50
Taxa de chegada estresse (TPS)	2.000	2.000
Número de células	10	10
Capacidade do pool de DB	50	50
Capacidade por célula	10	10
Latência log-normal média (ms)	50,0	50,0
Latência log-normal sigma	0,5	0,5
ACK event bus (ms)	2,0	2,0
Jitter ACK (log-normal sigma)	0,0	0,25
ACK std aditivo (ms)	0,0	0,3
Fator de penalidade de contenção	5,0	5,0
Probabilidade de cauda	0,0	0,003
Bin de análise de throughput (ms)	500	500

*Fonte: `experiment_provenance.json` — repositório GitHub (Flores, 2026).*